Maine State Government
Dept. of Administrative & Financial
Services Office of Information
Technology (OIT)

Software Development Lifecycle Procedure

I. **Statement**

Application Owners will use this Procedure to create the Software Development Lifecycle (SDLC) artifacts for major application projects, as required by the SDLC Policy[1].

II. **Purpose**

This SDLC Procedure is instituted in order to support the *SDLC Policy*. The *SDLC Policy* mentions artifacts to be created through the life of a major application project. This Procedure specifies the contents of such artifacts. However, beyond the contents, this Procedure does not take a position on the appearance and formatting of such artifacts.

III. **Applicability**

The applicability of this Procedure is identical to that of the *SDLC Policy*.

IV. **Responsibilities**

A. Application Owners: The Application Owners are responsible for creating the SDLC artifacts according to this Procedure and submitting them to the Enterprise Project Management Office (PMO).

B. Associate CIO, Applications: The Associate CIO, Applications, is responsible for enforcing this Procedure.

V. Guidelines & Procedures

A. Compliance with the *SDLC Policy* is measured by the successful creation of the SDLC artifacts according to this SDLC Procedure.

B. It is not the intent of the *SDLC Policy* and Procedure to create duplicate work. Should the Application Owners generate the contents of the SDLC artifacts under different names or contexts, then that adequately suffices in terms of compliance with the *SDLC Policy*.

[1] http://maine.gov/oit/policies/SDLCPolicy.htm

C. The artifacts, corresponding to each SDLC Discipline, are enumerated below. The artifacts are elaborated in the Definitions section, in alphabetical order.

Discipline	*Artifacts*
Business Modeling	• Current Process Flow Diagrams • Legacy User Manual *(Only if available, not mandatory.)* • Legacy Operations Manual *(Only if available, not mandatory.)*
Requirements	• Problem Statement • Desired Process Flow Diagrams • Desired Use Cases • Architectural Diagram
Analysis & Design	• Roles & Privileges Document • User Interface Prototype Document • Logical Database Design Document • Physical Database Design Document • Object-Relational Model Document (*Only if the Business Logic is encoded in an object-oriented programming language.)*
Test	• See the Application Deployment Certification Policy[2].
Implementation / Deployment	• Acceptance Plan • Infrastructure Planning Document • New Operations & User Manuals • Deployment Plan

D. The artifacts enumerated above constitute the minimum requirements in terms of compliance with the *SDLC Policy*. It is left to the discretion of the Application Owners to create additional SDLC documentation in support of their specific application projects.

E. Submission of the SDLC artifacts enumerated above does not preclude the necessity of submitting other project management documentation to the Enterprise PMO, concerning the overall management of the application project. The Enterprise PMO determines the scope and nature of the required project management documentation.

F. Alternative SDLC methodologies are known to use broad-based umbrella titles, such as Design Specifications, Requirements Document, etc. Contents of such titles may overlap with multiple artifacts enumerated above under Requirements and Analysis & Design. What actually matters in terms of compliance with the *SDLC Policy* is the content, not the title. Should it be more convenient to the Application Owners to generate the contents of the artifacts under Requirements and Analysis & Design under broad-based umbrella titles, then that adequately suffices in terms of compliance with the *SDLC Policy* and no duplication of effort is warranted. However, it still remains the burden of the Application Owners to prove that they have indeed generated all the required contents in their entirety, without exception, albeit under different titles.

[2] http://maine.gov/oit/policies/Application-Deployment-Certification.htm

VI. **Definitions**

1. Acceptance Plan: This artifact validates the successful deployment and formal acceptance by the Agency of the deployed solution.
2. Architectural Diagram: This artifact is a visual representation of the solutions architectural design.
3. Deployment Plan: This artifact is a detailed checklist used by the project team to ensure deployment readiness.
4. Infrastructure Planning Document: This artifact captures the provisioning blueprint for the infrastructure intended to stand up the application project. This includes complete details of all relevant client devices, terminal emulators, browsers, etc., as well as the full complement of web, application, and database servers.
5. Logical Database Design Document: This artifact is structured in terms of Entities and Relationships. While it is not expected to finalize all the Attributes of all the Entities, it is indeed expected to contain primary, alternative, and secondary keys, as well as their individual Domains. The Domain of an Attribute is defined by its data type, its size, and its permissible range of values.
6. Object-Relational Model Document: This artifact applies only if the Business Logic is encoded in an object-oriented programming language. It captures the mapping between objects in the memory with those on the disk. Most contemporary applications implement the Business Logic layer in an object-oriented programming language and house the Data layer in a database. This dichotomy necessitates a mapping between the two kinds of objects. At a minimum, this document should capture the correspondence between classes and tables. It is not mandatory to map inheritances and relationships. Reference [1] provides further guidance on this topic.
7. Operations Manual: This artifact captures all the instructions necessary for the operation and administration of the application, including the execution of batch jobs, restarting aborted/failed jobs, reviewing logs, and all weekly/quarterly/yearly periodic procedures.
8. Physical Database Design Document: This artifact is structured strictly in terms of the SQL Data Definition Language (DDL). In other words, this is the core of the 'Create & Alter Script' to be executed downstream by the DBA.
9. Problem Statement: This artifact captures the high-level summary of what is expected from the application project, as well as what is not expected. It may or may not define each and every deliverable. Nonetheless, all downstream deliverables should be in alignment with this Problem Statement, without exception. It should also include the background and context of the application project, including why it is being undertaken, and what business values it is meant to produce. This is also the only document that absolutely needs to be signed off by all Application Owners, without exception. Finally, should the contents of the Problem Statement be generated under the context of a Project Charter, then that

adequately suffices in terms of compliance with the *SDLC Policy* and no duplication of effort is warranted. However, it still remains the burden of the Application Owners to prove that they have indeed generated the entire contents of the Problem Statement, without exception, albeit under a different context.

10. Process Flow Diagram, also known as a Flow Chart: This is the most widely used process modeling tool and has been in continuous use for more than fifty years. Basic syntax includes squares for activities, rhomboids for input/output, diamonds for decision forks, arrows for control flow, and circles for off-page connectors.

11. Roles & Privileges Document: This artifact captures the various user/actor roles that are required in the application, as well as their access privileges to all the entities and transactions. The primary focus of this artifact should be at the business function level, rather than at the underlying database level. Thus, it is more critical to capture that a supervisor has the privilege to approve the timecards of their employees, and not the other way around, rather than establishing the Create-Read-Update-Delete Matrix for the timecard database table. Admittedly, this may not constitute a strict dichotomy in many cases and some prudent judgment needs to be exercised.

12. Use Case: A Use Case is a well-defined sequence of actions undertaken jointly by the user and the application, which produces a predictable result of value to the user. Use Cases are utilized to document business requirements and evaluate the users' ability to accomplish their work in a productive manner. Use Cases encompass input/output screens and forms, reports, tasks/transactions, interfaces, and batch jobs. At a minimum, a Use Case specification should include its name, version, summary, detailed primary scenario, detailed secondary scenarios, detailed exception scenarios, triggers, pre-conditions, post-conditions, assumptions, related business rules, code modules, authors, and last update date.

13. User Interface Prototype Document: This artifact captures the salient details of the input/output screens and forms, edit rules, report layouts and parameters, etc. While it is recognized that the actual implementations will include minutiae not covered by the prototypes, it is also expected that the actual implementations will not violate the salient details of these prototypes.

14. User Manual: This artifact captures all the instructions necessary for an average end-user to successfully interact with the application. More specifically, it covers data-entry, query, reporting, workflow, tasks/transactions, batch job schedules and their outcomes, etc. It does not cover operational and administrative aspects of the application.

VII. **References**

1. Agile Data, Mapping Objects to Relational Databases: O/R Mapping In Detail, Last Updated on October 6, 2006[3]

[3] http://www.agiledata.org/essays/mappingObjects.html

2. Software Development Lifecycle Policy[4]

VIII. **Document Information**

Initial Date: August 31, 2009
Last Revision Date: July 27, 2016

Point of Contact: Joshua Karstens, Director, Project Management Office, OIT, 207-624-8286.
Approved By: James R. Smith, Chief Information Officer, OIT, 207-624-7568.

Legal Citation: Title 5, Chapter 163: Office of Information Technology[5]
Waiver Process: See the Waiver Policy[6].

[4] http://maine.gov/oit/policies/SDLCPolicy.htm
[5] http://legislature.maine.gov/statutes/5/title5ch163sec0.html
[6] http://maine.gov/oit/policies/waiver.htm